Amendments to the Specification:

Page 1, line 25, to page 2, line 6, please amend the paragraph as follows:

One prior art method of having a server-free backup was to off load the control of the storage system backup to the Fibre Switches (see the white paper, "Deliver Server-Free Back-up," April 2000, Pathlight Technology, Inc., Ithaca, New York). Fig. 1 shows such a prior art system that uses a Fibre Channel Switch to perform the back-up after receiving a command from the host or server. Fig. 1 shows a server 110 coupled to its storage system, i.e., disk system 114, via a SAN having Fibre Channel Switch 112. A Tape Library 116 which is used for backup is also connected to Fibre Channel Switch 112 via a Fibre channel. The server 110 issues an Extended Copy (E-Copy) command 118 to a Fibre Channel Switch 112. The E-Copy is a SCSI Primary Command 2 or a vendor specific command, such as from Legato ® Systems, Inc. of Mountain View California (referred to herein as Legato ®) that instructs the copying of data from one logical device to another logical device. A copy manager in the Fibre Channel Switch 112 upon receiving the E-copy command from the Server 110 performs the Data transfer 120 by copying data from Disk System 114 to Tape Library 116. The copying proceeds under control of the Fibre Channel Switch 112 without need of server 110 action. Thus the server 112 is free to perform other tasks.

Page 3, lines 18-25, please amend the paragraph as follows:

In yet another embodiment of the present invention a storage system for executing an Extended Copy (E-Copy) command from a server is provided. The storage system is coupled with a plurality of back-up devices over a storage area network, including: a disk for storing data from the server[[;]]. There is a target port for receiving the E-Copy command, including a parameter list, where the parameter list lists the data for backup to a back-up device port; and there is an initiator port responsive to the E-Copy command for connecting to the back-up device port on the storage area network to backup the data to the backup device.

Page 6, line 26, to page 7, line 21, please amend the paragraphs as follows:

Each of the segment descriptors 450 uses the target descriptors 430 to specify a target descriptor source for the source of the data and a target descriptor destination for the

destination of the data. For example , let Target Descriptor 0 432 give the WWN for target port 222 with a LUN "A" referring to data block on disk unit 226 and let Target Descriptor "N" 434 refer to Backup device Port 240 and a LUN "B" related to a disk block on the Backup device. Segment Descriptor 0 452, in this example, would include information on the source: Target Descriptor 0 432 and designation of the backup: Target Descriptor N 434. It is then either manually or automatically up to the disk system to determine the initiator port(s) to be used to copy the data from LUN A on disk 226 to LUN B through port 240 on the Back-up device 232. In one embodiment a port group is set manually, for example, in this case, the group includes target port 222 and initiator port 224. In an alternative embodiment the disk system could in series check every initiator port (other initiator ports for Fig. 2 are not shown), until an initiator port is found that connects to Backup device Port 240 and a back up of LUN A is performed.

Fig. 5 shows another example of a parameter list of an embodiment of the present invention. Unfortunately, unlike the E-Copy command 404, there are ~~may~~ many vendor specific parameter list formats. The vendor specific format 460 of this example is of Legato ® . In comparing Legato ® 's parameter list to the SPC-2 parameter list 418: the list ID's 420 and 462, STR/NRCR/PRIORITY 422 and 466, target descriptor list length 424 and 464, and segment descriptor list length 428 and 468 are the same or similar in format. The 12^{th} byte 469 differs. In the SPC-2's parameter list 418 the 12^{th} byte is the inline data length 429, while in Legato's ® parameter list 460, there is no inline data size (or inline data), but the byte 12^{th} is the target descriptor 0 first byte 469, e.g., "E0h," (in contrast for SPC-2, the first byte, also "E0h," occurs at the 16^{th} byte 430 in Fig. 4). Since the Inline Data Length 429 is 4 bytes long, presetting it to "00h" should have only a small effect on the maximum amount of Inline Data 455 is allowed in Fig. 4. Thus if byte 12 is checked in both the SPC-2 and Legato ® formats, it is "00h" for SPC-2 and "E0h" for Legato ®, the system can distinguish between the two the parameter list formats. Therefore this embodiment can be used for both the Legato ® and SPC-2 formats.

Page 9, line 27, to page 10, line 24, please amend the paragraph as follows:

Fig. 8 shows an example of groups of target and initiator ports for a disk system of one embodiment of the present invention. In Fig. 8 there ~~is~~ are a plurality of servers, for example, server A 712 server B 714, server C 740, and server D 742. The plurality of servers are

connected to a disk system 710, which is also connected to a plurality of backup device ports, for example, backup device A 736, backup device B 738, backup device C 758, and backup device D 760, via one or more SAN's. For the purposes of illustration the individual back-up ports are not shown, although it should be understood that the connection is to the backup port not the device. Also SAN's 716, 744, 734, and 756 may all be one big SAN or the may be a combination of one or more storage area networks. The purpose of the disk system 710 in Fig. 8 is to receive E- copy commands from a plurality of servers and back up the appropriated disk information associated with those servers to the available backup devices. The purpose of the groups of ports or port groups, for example, Group 1 730 consisting of target port A 720, target port B 722 and initiator port E 732 and, for example, the group 2 750 consisting of target port D 746, and initiator port G 752, and initiator port H 754, is to use the users knowledge of the system in assigning target ports and initiator ports in order to improve the efficiency and security of locating the backup device ports. In an alternative embodiment the disk system 720 could search all the initiator ports to locate the appropriate backup device port, when a back-up from a target port is requested. In the example in Fig. 8, server A 712 is connected via SAN 716 to target port A 720 and server B 714 is connected via SAN 716 to target port B 722 (or in the alternative, the connections can be switched). The initiator Port 732 of the disk system 710 locates through SAN 734, for example, a port on device A 736 as the target backup device port. In the example of Group 2 750, either server C 740 or server D 742 can connect via SAN 744 to target port D 746. Then initiator port G 752 or initiator port H 754 may be used for backup. Initiator port G 752 may be connected to device D 760 through SAN 756 and initiator port H 754 may be connected to a port on device C 758 and also to the above target port on device D 760. The numbers and configurations of the servers and backup devices and the number and combinations of target ports and initiator ports and groups in disk system 710 are for illustration purposes only and are not meant as a limitation on the scope of the invention.

Page 11, lines 3-30, please amend the paragraph as follows:

Fig. 9 illustrates a simplified example of a backup procedure for an embodiment of the present intention. The server 808 sends an E-Copy command 810 to port A (target port) 812 of Disk system 805. The E-copy command starts the execution of target JOB 816 on

microprocessor 814. Target JOB 816, using the parameter list of the E-copy command, then places in shared memory (SM) 820 a bitmap table 818 indicating which microprocessors may be used to execute ~~to~~ the E-copy command, i.e., the backup of server 808 data stored on a logical unit on a disk (not shown) to a tape device port 880. For illustration purposes assume both microprocessors 830 and 832 are in the bitmap table 818. Microprocessors 830 with initiator port B 840 and microprocessor 832 with initiator port C 842 run concurrently with microprocessor 814. Both microprocessors 840 and 842 kernels concurrently poll the SM 820 to determine if they can start the E-Copy job (836 and 838). In this example let microprocessor 830 access the bitmap 818 first, seize exclusive control of the SM 820, and assume it determines it may start the E-copy job. The microprocessor 830 through the port B 840 then searches 850 for tape device port 880. In this example of the search 850, microprocessor 830 (via initiator port B 840) cannot connect to tape device port 880. The reasons for the non-connection may include, tape device port 880 is not available or there is no available SAN connection or tape port 880 is busy. Next, the microprocessor 830 releases its control of SM 820. Assume next that microprocessor 832 takes exclusive control of SM 820 and by examining the bitmap table 818 determines it may start an E-copy job 838. The microprocessor 832 through initiator port C 842 again searches for tape device port 880 and finds it available. The microprocessor 832 then executes the data transfer 852 from the logical unit in the disk system 805 through initiator port C 842 to tape device port 880. Upon completion of the backup, microprocessor 832 notifies the Target job 816 on microprocessor 814 that it has successfully completed the backup, and microprocessor 814 then notifies the server 808 via target port A 812, that the backup is complete. If microprocessor 843 also cannot connect to tape device port 880, then an error message is sent back to the server 808 via microprocessor 814.

Page 12, line 18, to page 13, line 18, please amend the paragraph as follows:

Fig. 11 shows a flowchart for the processing on the initiator port side of a disk system of an the embodiment of the present invention. From step 930 in Fig. 10 (from continuation 1), at step 1010, the initiator port microprocessors concurrently poll the shared memory to check if the target job has put the bitmap table in Shared Memory (SM), e.g., 650. At step 1012 one microprocessor[[s]] that is polling the SM discovers there is a the bitmap table

placed in SM by the target JOB, and that μP takes exclusive control of the SM and checks the bitmap table. At step 1014 the kernel tests if it can start the E-copy job on its microprocessor. If the kernel cannot start the E-copy job, for example, the microprocessor is not in the bitmap table, then the kernel checks the bitmap table again at step 1016. At step 1018, if there are other microprocessors that may be able to start the E-copy job, i.e., there are other unchecked microprocessors in the bitmap table, then the present kernel releases the shared memory at step 1020 and another polling kernel takes exclusive control of the shared memory at step 1012 and the process repeats. At step 1030, if the kernel can start the E-copy job on its microprocessor, the E-copy job starts. At step 1032 the E-copy job gets the parameter list for the E-copy command, for example, table 418 in Fig. 4, from the shared memory. At step 1034 the E-copy job searches for the backup device target port given in the parameter list for each initiator port assigned to the microprocessor. For example, microprocessor 642A is connected to both port[[s]] 622 and port 624. Hence μP 642A searches, for example, for the back-up device port, for example Backup Device Port 610, on initiator port 622 first. Then finding no available back-up port, μP 642A next searches for the back-up device port given in the parameter list, on initiator port 624. At step 1036 the kernel tests if the E-copy job found the available backup device port for the E-copy. If a backup device port was not found or not available, then at step 1038, the microprocessor is marked unavailable in the bitmap table and at step 1018 the kernel checks the bitmap table again to see if there are other microprocessors to start the E-copy job. If there are none, then at 940 (to continuation 3), the process returns to step 942 in Fig. 10. If the initiator port can find the backup device port for the E-copy JOB, then at step 1040 the E-copy job starts the data transfer from the logical device to the backup device. When the data transfer is finished, the E-copy job is finished (step 1042). The bitmap is removed from the SM. The kernel sends the results of the E-copy job processing to the target job at step 1044 and proceeds to 940 (continuation 2) in Fig. 10; and also the kernel proceeds to step 1010 and waits for the next bitmap table to be placed in the shared memory by another target JOB.

Page 15, lines 6-19, please amend the paragraph as follows:

Fig. 12 shows a GUI for changing target/initiator port designations on a disk system of an embodiment of the present invention. The display screen 1110 shows a list of ports

with associated target or initiator port designations 1112, for example CH A(CL1) Target 1114, CH B(CL1) Target 1116, CH A(CL2) Target 1120, and CH C(CL1) Initiator 1122. The terms CL1 and CL2 refer to the redundant disk controller systems , 605A and 605B, respectively. Since the disk controller systems have redundant ports, the terms CL1 and CL2 are used to distinguish between the disk controller system ports. A port is specified on the screen 1112 and changed to an initiator port by clicking on the initiator button 1130 or changed to **[[an]]** a target port by clicking on the target button 1132. For example CH A(CL2) Target port 1120 can be changed to CH A(CL2) Initiator port by highlighting CH A(CL2) Target 1120 on area 1112 and then clicking on the initiator 1130 button. The results of the port designations may be shown on a display with images similar to Fig. 8 or on a display showing only an image of the disk device 710 with associated ports.